4. Writing to a File

Write a Ruby function `merge_json_files` that merges JSON objects from one file into another.

file : file.json

- Function prototype:`merge_json_files(file1_path, file2_path)`

```
┌──(imen㉿hbtn-lab)-[…/scripting_cyber/0x00-ruby_scripting]
└─$ cat 4-main.rb
require_relative '3-read_file'
require_relative '4-write_file'

file1_path = 'file_to_copy_from.json'
file2_path = 'file.json'


merge_json_files(file1_path, file2_path)


puts 'File merged successfully!'

count_user_ids(file2_path)
```

```
┌──(imen㉿hbtn-lab)-[…/scripting_cyber/0x00-ruby_scripting]
└─$ ruby 4-main.rb
1: 10
2: 8
3: 11
4: 13
5: 7
6: 4
7: 10
8: 8
9: 2
10: 1
11: 2
12: 1
13: 4
```

```ruby
require 'json'

def merge_json_files(file1_path, file2_path)
  # Read and parse the first JSON file
  file1_content = File.read(file1_path)
  data1 = JSON.parse(file1_content)

  # Read and parse the second JSON file
  file2_content = File.read(file2_path)
  data2 = JSON.parse(file2_content)

  # Merge the two JSON objects (append data1 to data2)
  merged_data = data2 + data1

  # Write the merged data back to file2
  File.write(file2_path, JSON.pretty_generate(merged_data))

  puts 'Merged JSON written to file.json'
end
```

How It Works

1. **Read and Parse Both Files:**
   - **`File.read(file1_path)`**: Reads the content of the first JSON file.
   - **`JSON.parse(content)`**: Converts JSON strings into Ruby objects (arrays of hashes).
2. **Merge the Data:**
   - `data2 + data1`: Appends the objects from `file1` to `file2`.
3. **Write Merged Data:**
   - **`JSON.pretty_generate`**: Converts the merged Ruby object back into a formatted JSON string.
   - **`File.write(file2_path, ...)`**: Writes the merged JSON back to `file2_path`.
4. **Output:**
   - Once the files are merged, the `count_user_ids` function is called to count and display the `userId` occurrences in the updated `file.json`.